AMENDMENTS TO THE SPECIFICATION

Please replace the heading on page 1, line 11 with the following amended heading:

~~BACKGORUND~~ BACKGROUND OF ART

Please replace paragraph [0007] with the following amended paragraph:

[0007]

Methods for solving part of the above-mentioned problems partially are also proposed. For example, in nonpatent literatures No. 3 (A. Buldas, P. Land, H Lipmaa and ~~B. Schoenmakers, Optimally efficient accountable time-stamping, in Proceedings of Public Key Crytography 2000 (PKC2000), eds. Y Zheng and H. Imai, pp. 293-305, Springer-Verlag, January 2000)~~ J. Villemson: Time-stamping with binary linking schemes, in Processings of Advances on Cryptology (CRYPTO'98), ed. H. Krawczyk, pp.486-501, Springer-Verlag, 1998) and No. 4 (A. Buldas, H Lipmaa and B. Schoenmakers, Optimally efficient accountable time-stamping, in Proceedings of Public Key Crytography 2000 (PKC2000), eds. Y Zheng and H. Imai, pp. 293-305, Springer-Verlag, January 2000), there is proposed a method of adopting a tree structure in place of the linear lists used in the nonpatent literatures Nos. 1 and 2, in order to calculate publication data collecting up event-ordering requests processed by an event-ordering certification apparatus for a certain period, thereby remarkably reducing the amount of data required for the user 30 to verify an event-ordering receipt, from the amount of data proportional to the number of event-ordering requests accepted for the certain period to the amount of data proportional to a logarithm (base 2) of the former amount.

Please replace paragraph [0028] with the following amended paragraph:

[0028]

We now describe an example of a sequential aggregation tree having sixteen leaves, as shown in Fig. 5. The number of leaves of the sequential aggregation tree and its height do not become definite unless the sequential aggregation period is completed. Further, in the sequential aggregation tree, the assignment of values to the leaves is carried out from left, in sequence. The assignments of values to nodes higher than level 0 (i.e. non-leaves) are carried out incrementally if possible. Accordingly, for a plurality of nodes on the same vertical line of Fig. ~~3~~ 5, the assignments of values to the nodes are carried out at about the same time in the same processing unit. Under the notation that a node at level j and numbered (index) i is represented by (j, i) and an assigned value of (j, i) is represented by V(j, i), the concrete example of Fig. 5 will be described.

Please replace paragraph [0043] with the following amended paragraph:

[0043]

The format of an audit receipt is identical to that of a receipt to be sent to the user apparatus 2i. Note that digital data y as a basis of calculating a sequentially assigned data-item may be data that was sent from the audit apparatus 3 to the certification apparatus 1, as an audit request. Alternatively, the digital data y may be produced by the relevant certification apparatus 1 in accordance with a predetermined procedure for the audit apparatus ~~2~~ 3 in question. Additionally, on the assumption of drafting a digital document as an object of event-ordering certification in the audit receipt in accordance with a predetermined procedure, a hash value as a result of applying a predetermined hash function on the digital document may be adopt as the digital data for calculating the sequentially assigned data-item.

Please replace paragraph [0050] with the following amended paragraph:

[0050]

Here, we now describe the function of the event-ordering certification ~~verifying~~ audit part 32 with reference to Fig. 5. In Fig. 5, because of an audit point (0, 10), the audit information that the audit apparatus 3 receives at this point of time comprises V(3, 0) and V(1, 4) as mentioned above. On the other hand, the user apparatus 2i sends, as the audit request information, V(0, 5) and V(0, 4), V(1, 3) and V(2, 0) as the sequential aggregation complementary data. Regarding the incorporation of V(1, 3) into the audit request information, it is noted that it becomes possible for the user apparatus 2i to acquire V(1, 3) (not included in the immediate complementary data) from the certification apparatus 1 at the point of requesting the validation (i.e. on and after the audit point (0, 10) temporally behind the issue of the point (0, 5)) and therefore, the user apparatus 2i actually acquires V(1, 3) as the late complementary data from the certification apparatus 1 and incorporates it into the audit request information. In this way, the event-ordering certification ~~verifying~~ audit part 32 verifies whether its own audit information V(3, 0) coincides with V(3, 0) introduced by the audit request information from the user apparatus 2i.

Please replace paragraph [0061] with the following amended paragraph:

[0061]

Next, the event-ordering request aggregation part 12 assigns a sequentially assigned data-item for audit calculated by the audit event-ordering request to a sequential aggregation-tree leaf to construct a sequential aggregation-tree incrementally. While, the audit-information drafting part ~~13~~ 14 drafts a receipt certificate for audit (referred to as "audit receipt" after) and successively sends it to the audit apparatus 3 through the transmitting/receiving part 11 (steps S90, S100, S110).

Please replace paragraph [0074] with the following amended paragraph:

[0074]

The audit apparatus 3 receives the audit request information through the transmitting/receiving part 31 (step S470). In respective leaves of a sequential aggregation tree where audit receipts on previous reception are assigned, the event-ordering certification audit part 32 calculates an audit point α between a "sequential aggregation tree" leaf τ that the receipt in the above audit request information on this reception is assigned and a leaf τ' included in the late complementary information (step S480). Next, the audit apparatus 3 calculates a certification point for the leaf τ by the audit point α from the audit request information and further calculates an assigned value "Acal" for the so-calculated certification point (step S490). On the other hand, the event-ordering certification ~~verifying~~ audit part 32 acquires an assigned value A of this certification point that the apparatus 3 has already acquired as the audit information, from the memory part 33 and judges whether the assigned value A of the certification point coincides with the assigned value "Acal" of the certification point on calculation (steps S500, S510).

Please replace paragraph [0114] with the following amended paragraph:

[0114]

All one can firstly say from this validation result is as follows. Let t1, t2 and t2' denote a time when the audit event-ordering request of the audit apparatus ~~7~~ 8 corresponding to the audit point α1 is received by the certification apparatus 7, a time when the event-ordering request of the user apparatus 2i corresponding to the user point τ is received by the certification apparatus 7 and a time when the receipt against the event-ordering request is

transmitted from the certification apparatus 7, respectively. Then, the inequality t1 < t2' is satisfied.

Please replace paragraph [0116] with the following amended paragraph:

[0116]

It should be noted that the above argument couldn't be effected unless the user apparatus 2i receives the instant complementary data forming the receipt despite that the serialisablity of the certification apparatus ~~1~~ 7 is ensured until the time t2'. Because it is impossible to eliminate a possibility that the certification apparatus 7 changes an assigned value at α1 after the time t2'.

Please replace paragraph [0132] with the following amended paragraph:

[0132]

In the following descriptions about the sequential aggregation tree leaves τ and τ1, a terminology "time point τ (or τ1)" represents a point of time when an event-ordering request assigned to τ (or τ1) is received. Therefore, $\tau \leqq \tau$1 represents that the time point τ1 is present after the time point τ. Assume in the following descriptions τ2 represents a larger one in τ and τ1, and additionally, the serialisablity of the certification apparatus ~~1~~ 7 is ensured until its transmission of a receipt against an event-ordering request received at the leaf τ2.

Please replace paragraph [0163] with the following amended paragraph:

[0163]

The user time-stamping apparatus 20i has the function of a time-stamping apparatus in addition to the function of the user apparatus 2i, as mentioned above. The user time-stamping apparatus 20i comprises a transmitting/receiving part 21 for transferring data to and from the audit apparatus 9, the user apparatus 10j and the time information offering apparatus 90, a time stamp drafting part 201 for drafting a time receipt on acceptance of a time-stamping request from the user apparatus 10j, an event-ordering requesting part 202 for requesting a certification containing a time receipt digest, a complementary data requesting part 23 for requesting complementary data of a receipt, which is acquirable at the present moment, an event-ordering certification verifying part ~~204~~ 203 for verifying the receipt and a memory part ~~205~~ 204 for storing information about event-ordering certification including the receipt and information about time-stamping including the time receipt. Note that although this embodiment adopts a user apparatus doubling as a time-stamping apparatus, there may exists a user apparatus that does not double as the time-stamping apparatus, allowing provision of a system structure where the user time-stamping apparatuses 20 and the user apparatuses 2i are mixed together.

Please replace paragraph [0164] with the following amended paragraph:

[0164]

In detail, the time stamp drafting part ~~202~~ 201 accepts the time-stamping request including designated digital data transmitted from the user apparatus 10j and successively drafts the time receipt where the time information from the time information offering apparatus 90 is attached to the digital data.

Please replace paragraph [0165] with the following amended paragraph:

[0165]

The event-ordering requesting part ~~203~~ 202 operates to incorporate the time receipt digest (i.e. a hash value of the time receipt drafted for the time-stamping request from the user apparatus 10j) into an event-ordering ~~receipt certification~~ request. In detail, this time receipt digest corresponds to a result of applying a "collision-resistant" one-way hash function, which is prepared by the user time-stamping apparatus 20i in advance, on the time receipt. Accordingly, the receipt that the user time-stamping apparatus 20i receives from the certification apparatus 1 has a structure shown in Fig. 6. However, as mentioned above, the original digital data y in the certificate contains the time receipt digest.

Please replace paragraph [0167] with the following amended paragraph:

[0167]

The block-time-stamping certificate drafting part 91 acquires the time of its receiving an audit receipt from the certification apparatus 1, from the time information offering apparatus 30 and further attaches the time to the block-time certificate. Thus, in this embodiment, the block-time certificate drafted by the block-time-stamping certificate drafting part 91 includes a time stamp bounding on the future side. As previously mentioned in the first embodiment, since the validation of an event-ordering certificate using the audit apparatus 3 (i.e. the second validation by the user apparatus 2i) makes it possible to certify that the leaf of the sequential aggregation tree where the event-ordering ~~receipt certification~~ request is assigned is temporally former of the leaf of the audit point, the time stamp bounding on the future side certifies nothing but the acceptance of a time stamping request from the user apparatus 10i having its origin in requesting the event-ordering certificate is temporally former of the time when the audit apparatus 9 received the audit receipt. This block-time certificate in this embodiment will be referred to as "the first-class block-time certificate" after.

Please replace paragraph [0168] with the following amended paragraph:

[0168]

Note that the above apparatuses are formed by electronic apparatuses each having a CPU (Central Processing Unit) having at least a calculating function and a control function, a main memory having a function to store programs and data, such as RAM (Random Access Memory), and a secondary memory capable of continuing to memorize data even at powered-off, such as HD (Hard Disc). The operations of respective parts of the user time-stamping apparatus 20i (i.e. the time stamp drafting part ~~202~~ 201, the event-ordering requesting part ~~203~~ 202, the complementary data requesting part ~~204~~ 23, the event-ordering certification verifying part ~~205~~ 203) and the operation of the block-time-stamping certificate drafting part 91 of the audit apparatus 9 are nothing but respective crystallizations of the above calculating/control functions of the above central processing unit. Additionally, the memory part ~~206~~ 204 of the user time-stamping apparatus 20i and the memory part 92 of the audit apparatus 9 are respectively equipped with the above-mentioned functions of either the main memory or the secondary memory.

Please replace paragraph [0171] with the following amended paragraph:

[0171]

Regarding the event-ordering certification method, its overall operation is substantially the same as the operation of Fig. 8, assuming that the user time-stamping apparatuses 20i and the audit apparatus 9 correspond to the user apparatuses 2i and the audit apparatus 3, respectively. Therefore, the following descriptions are mainly directed to an interaction between the user time-stamping apparatus 20i and the user apparatus 10j, which is different from the operation of Fig. 8. Figs. ~~20~~ 22 and 23 are sequence diagrams to closely explain the operation of step S10' to send an event-ordering request, corresponding to step

S10. Note that Fig. 22 also contains step S60' for receiving the receipt, corresponding to step S60 of Fig. 8. Further, Fig. 24 is a sequence diagram to closely explain the operation of step S12' to receive a receipt certificate for audit (referred to as "audit receipt" later), corresponding to step S120 of Fig. 8.

Please replace paragraph [0174] with the following amended paragraph:

[0174]

When the user apparatus 10j sends a time stamping request including digital data to the user time-stamping apparatus 20i, it receives the time stamping request including digital data through the transmitting/receiving part ~~201~~ 21 (steps S11', S12'). Next, the time stamp drafting part 201 of the apparatus 20i acquires the time of receiving the time stamping request from the time information offering apparatus 90, drafts a time receipt certificate (referred to as "time receipt" after) having the time applied on the digital data and send the time receipt to the user apparatus 10j (steps S13', S14', S15). In this way, the user apparatus 10j can acquire the time receipt (step S20').

Please replace paragraph [0175] with the following amended paragraph:

[0175]

Next, the event-ordering requesting part ~~203~~ 202 of the user time-stamping apparatus 20i drafts a digest of the time receipt, further drafts an event-ordering request including this "time receipt" digest and sends it to the certification apparatus 1 (steps S17', S18'). In this way, the certification apparatus 1 receives the event-ordering request through the transmitting/receiving part 11 (step S20').

Please replace paragraph [0202] with the following amended paragraph:

[0202]

Let [i1 ..i2] be the set (interval) of integers i satisfying i1 ≦ i ≦ i2 for two integers i1 and i2; ~~(i1 ..i2)~~ [i1 ..i2] be the set (interval) of integers i satisfying i1 < i ≦ i2; (i1 ..i2) be the set (interval) of integers i satisfying i1 ≦ i < i2; and ~~(i1 ..i2)~~ [i1 ..i2] be the set (interval) of integers i satisfying i1 < i < i2.

Please replace paragraph [0204] with the following amended paragraph:

[0204]

Assume that the number of event-ordering requests to be accepted for an aggregation period (e.g. one week) is previously fixed by a method of some kind. Let n be the fixed number of requests. Then, the height of the aggregation tree is k = ceiling($\log_2$(~~N~~ n)). Here, the maximum number of leaves in the tree of height k is 2^k. Thus, on condition d=2^k – n, if only eliminating nodes at level 0 in the number of 2d, then it becomes possible to assign the event-ordering requests (number: n) to respective leaves without producing any dummy node. The reason is as follows: If the number of leaves at level 0 is reduced by number 2d, then new leaves at level 1 (number: n) are produced. As a result, due to a reduction in the number of leaves by number d, the total number of leaves results in n = 2^k – d.

Please replace paragraph [0206] with the following amended paragraph:

[0206]

place(i) = (0, i) (0 ≦ ~~I~~ i < L0L),

place(i) = (1, L1L + i – L0L) (L0L < i ≦ n)

where place(i) = (level, number).

Please replace paragraph [0207] with the following amended paragraph:

[0207]

Fig. 26 shows a concrete example of the first method of dynamically forming the sequential aggregation tree in case of n = 10. In this case, as shown in Fig. 26, there is established k = ceiling($\log_2(10)$) = 4, and therefore the height becomes 4. Then, as d = ~~24~~ 2^4 – 10 = 6, the leaves at level 0 in the number of 12 = 6×6 are deleted. As a result, L1W = ~~23~~ 2^3 = 8, L1L = 8 – 6 = 2, L0L = 2×2. The numbers of leaves are: 4 leaves at level 0; 6 leaves at level 1; and total number n = 10. Consequently, the aggregation tree shown in Fig. 24 can be formed dynamically. When possible, it is carried out to assign values to respective nodes whose levels are more than 0 incrementally.

Please replace paragraph [0250] with the following amended paragraph:

[0250]

Additionally, assume that authPathDV(p, m) and authPathTDV(p, m) represents authPathD(~~p~~ t, m) and authPathTD(t, m) plus assigned values of respective nodes belonging to authPathD(p, m) and authPathTD(p, m), respectively. In detail, when authPathDV(p, m) and authPathTDV(p, m) are expressed as above, there are established:

authPathDV(p, m) =

[((j(0), a(j(0)), v(j(0)), …, ((j(k1 –1), a(j(k1 –1))), v(j(k1 –1)))] , and

authPathTDV(p, m) =

[((j(0), a(j(0)), LR(j(0)), v(j(0)), …,

((j(k1 –1), a(j(k1 –1))), LR(j(k1 –1)), v(j(k1 –1)))]

where v'(j) = V(j, a(j')) for each j' ∈ {j(0), …, j(k1-1)}.

Please replace paragraph [0258] with the following amended paragraph:

[0258]

By this procedure, it is carried out to calculate both root path rtPath((0, i0), m) for node (0, i0) and root path rtPath((0, i1), m) for node (0, i1). As a result, rtPath((0, i0), m) comes to coincide with rtPath((0, i1), m) since a certain element. Then, the element where rtPath((0, i0), m) coincides with rtPath((0, i1), m) at first is called "confluent point" between node (0, i0) and node (0, ~~i0~~ i1). Further, a left child of the confluent point is referred to as "authentication point of node (0, i0) (i.e. user point) by node (0, i1) (i.e. audit point)".

Please replace paragraph [0276] with the following amended paragraph:

[0276]

(Case 1) When p2 is the right child of p3, an assigned value V(p2) for p2 is included in the late complementary data CToken(i0, i2) that the apparatus C can receive at i2 satisfying i1 ≦ j2, as shown in Fig. ~~30~~ 32. The reason is as follows. By the feature 1 of the sequential aggregation tree, when the event-ordering certification process on the round corresponding to leaf (0, i1) is completed, it has already become possible to calculate an assigned value for a partial tree of curSBT(i1) indicated with B of Fig. 32. As a matter of fact, the assigned values have been already calculated and assigned. Accordingly, the late complementary data published on and after the above point of completion contains the assigned value V(p2) for the root p2 of the partial tree B.

Please replace paragraph [0314] with the following amended paragraph:

[0314]

The certification apparatus 1a comprises a transmitting/receiving part 11a for transmitting and receiving data to and from the user apparatuses 2I through the computer network 3a, an event-ordering request aggregation part 12a for arranging digital data (as event-ordering requests) transmitted from the user apparatuses 2I with the use of a sequential aggregation tree, an event-ordering reply drafting part 13a for drafting a certification reply containing the receipt, a digital signature drafting part 14a for applying a high-intensity digital signature on data where respective contents of a plurality of receipts published for a constant period by the certification apparatus 1a are aggregated, thereby forming publication data, an electronic information publishing part 15a for publishing the publication data electronically and a memory part 16a for storing the receipts and information about event-ordering certification.

Please replace paragraph [0393] with the following amended paragraph:

[0393]

Next, for a node at level more than 1 whose assigned value becomes calculable as a result of the addition of the new registration point, the certification apparatus 4a calculates the assigned value and stores the positional information of the node and the assigned value in the memory part ~~41a~~ 42a (step S1107a).

Please replace paragraph [0394] with the following amended paragraph:

[0394]

Next, in accordance with the definition of late complementary data, the certification apparatus 4a acquires late complementary data of an immediately-preceding registration

point from the immediately-preceding registration point with respect to each user apparatus 5I and the sequential aggregation tree stored in the memory part ~~41a~~ 42a (step S1109a).

Please replace paragraph [0399] with the following amended paragraph:

[0399]

The operation of the certification apparatus 4a in the method B will be described with reference to Fig. 49. Fig. 49 is a flow chart showing the functions of the event-ordering request aggregation part 12a and the event-ordering reply drafting part 41a.

Please replace paragraph [0451] with the following amended paragraph:

[0451]

At step ST1215a, it is executed to set 1 to the local variable lev_xb, V(1, 4) to val_xb and set 5(= 1+4) to idx_1b. Further, (lev_xb, idx_1b) = (1, 5) is set to place_1b. This positional information (1, 5) represents a first dummy node. Then, it is executed to call out a function "dummy_hash" to calculate a hash value to be assigned to the first dummy node and further executed to set its return value to "dum_val_1b". Additionally, it is executed to set "sfml_1b" as

make-stackflm(place_1b, dum_val_1b) = [(1, 5), dum_val_1b] .

Here, "make-stackflm" is a function of producing a stack frame while setting the positional information about node and its hash value as arguments. Here, while setting ~~plac_1b~~ place_1b and sfml_1b as arguments, there is called out REGISTER_COMPLE_DATA (defined with Fig. 53).

Please replace paragraph [0452] with the following amended paragraph:

[0452]

At next step S1216a, it is executed to set 2 to "lev_nw", floor(idx_xb, 2) = floor(4, 2) = 2 to "idex_nw" and set (2, 2) to "place_nw". In succession, it is executed to set hash_comb2(val_xb, dum_val_1b) to "val_nw". On establishment of two hash values as arguments, "hash_comb2" is a function of returning a result of applying a designated hash function to a junction between these hash values. Setting "make_stackflm(place_nw, val_nw)" to "sflm_nw", "~~sfml~~ sflm _nw" is pushed against "_stack". Consequently, "_ stack" has a structure including [(2, 2), V(2, 2)] and [(3, 0), V(3, 0)]. Further setting both place_nw and ~~sfml~~ sflm _nw as arguments, there is called out REGISTER_COMPLE_DATA. Then, the routine is returned to step S1211a.

Please replace paragraph [0458] with the following amended paragraph:

[0458]

Next, at step ST1211a, it is executed to set 4 to "lev_nw", floor(1, 2) = 0 to "idex_nw" and set (4, 0) to "place_nw". In succession, it is executed to set hash_comb2(V(3, 0), V(3, 1)) to "val_nw". This value is represented by V(4, 1). Setting "make_stackflm((4, 0), V(4, 0))" to "sfml_nw", "sfml_nw" is pushed against "_stack". Consequently, "_ stack" has a structure including [(4, 0), V(4, 0)]. Further setting both place_nw and ~~sfml~~ sflm _nw as arguments, there is called out REGISTER_COMPLE_ DATA. Then, the routine is returned to step S1211a.

Please replace paragraph [0470] with the following amended paragraph:

[0470]

(Incremental Individual Completion)

Assume that a registration point af is a provisional terminal point that belongs a certain aggregation interval I of the user apparatus ~~2A~~ 5A or coincides with a first registration point in the next aggregation interval.

Please replace paragraph [0493] with the following amended paragraph:

[0493]

Fig. 62 shows a calculation procedure FOREST_SST for determining the sequential aggregation small tree ST. This routine corresponds to step S520a of Fig. ~~62~~ 60.

Please replace paragraph [0495] with the following amended paragraph:

[0495]

• as inputs, a leaf identifier a (nonnegative integer) and an identifier "fin" (nonnegative integer) of provisional terminal point;

• as outputs, a leftmost leaf identifier "start" (nonnegative integer) in a sequential aggregation small tree containing a and a ~~leftmost~~ rightmost leaf identifier "last" (nonnegative integer) in the sequential aggregation small tree containing a;

• as variables, respective variables "rest", "ht" and "leaf_rum" for retaining nonnegative integers; and

• as usable functions, $\log_2(x)$: a maximum integer less than $\log_2(x)$; expt(x, y): x^y.

Inputting the leaf identifier a (nonnegative integer) and the identifier "fin" (nonnegative integer) of the provisional terminal point and further assuming that "ST" represents a sequential aggregation small tree containing a and also belonging to the sequential aggregation tree at the point of completing the registration of the provisional terminal point, this algorithm outputs the leftmost leaf identifier "start" (nonnegative integer) and the rightmost leaf identifier "last" (nonnegative integer) in pairs. The number of leaves

in the relevant sequential aggregation small tree amounts to "last – start + 1" and the height of the relevant sequential aggregation small tree becomes $\log_2$(last – start + 1).

Please replace paragraph [0507] with the following amended paragraph:

[0507]

Further, the authentication path of "a0" in the small tree ST is represented by authPathST(a0) where

authPathST(a0) = [(0, s(0)), (1, s(1)), …, (k–1, s(k –1))].

(Note that k is a height of the sequential aggregation small tree ST that "a0" in the completed forest belongs to. That is, k = height(ST).) In the following descriptions, for nonnegative integers n and m, it is assumed that [n…m] represents an aggregate of integers more than n and less than m.

Please replace paragraph [0509] with the following amended paragraph:

[0509]

Let "rtPathST(a0) ST" denote a root path of "a0" in the tree ST, where

rtPathST(a0) = [(0, r(0)), (1, r(1)), …, (k–1, r(k –1)), (k, r(k))],

r(0) = a0, and root(ST) = (k, r(k)). Assume that j∈[0…k–1].

Please replace paragraph [0531] with the following amended paragraph:

[0531]

(3) In accordance with the above procedure DECIDE_GET_POINT_A described with Fig. 63, it is executed to determine an acquisitive reference point a2 of V(j, a(j)) (step

S5505a). Note that "a2" is one of registration points, which allows a calculating of V(j, a(j)) from the chain complementary data acquired at the one registration point.

Please replace paragraph [0572] with the following amended paragraph:

[0572]

In a concrete example of Fig. 74, when completing certificates of registration points in the thinned-out extraction data numbered indexes 0, 1 and 2, in other words, the registration points of leaf numbers 1, 11, and ~~3~~ 31, it is possible to calculate assigned values of all of the authentication path nodes of three registration points. For instance, as for one registration point in the thinned-out extraction data numbered index 0, it is possible to calculated assigned values for nodes (0, 0), (1, 1), (2, 1), (3, 1), and (4, 1).

Please replace paragraph [0631] with the following amended paragraph:

[0631]

(Case 1-1) When p2 is the right child of p3, an assigned value V(p2) of p2 is included in the late complementary data lateData(i0, i2) that the apparatus C can receive at i2 satisfying i1 ≦ j2, as shown in Fig. 82. The reason is that when the event-ordering certification on the round corresponding to leaf (0, i1) is completed, it has already become possible to calculate an assigned value for an "S~~B~~T2" partial tree indicated with B of Fig. 82. As a matter of fact, the assigned values have been already calculated and assigned. Accordingly, the late complementary data for the registration point i0 published on and after the above point of completion contains the assigned value V(p2) for the root p2 of the partial tree B.